# **Simple Processor**

#### **Author**

Kirk Xu

Patrick Peng

Yuhan Xue

#### Work

We implemented some basic functionalities of a processor. Only implemented the following R-type and I-type instructions in this project:

add, addi, sub, and, or, sll, sra, sw, and lw.

## **Standard**

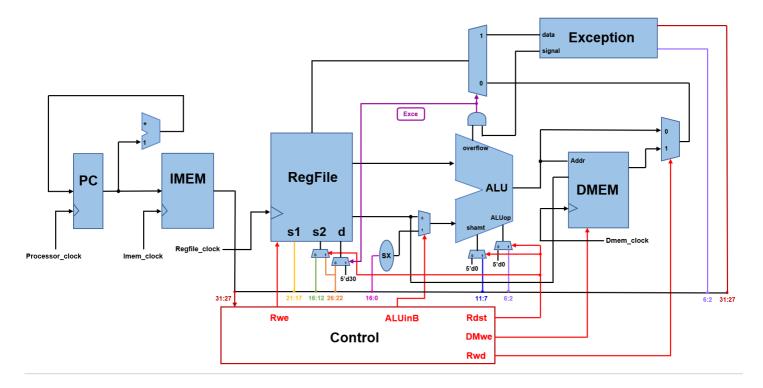
The instruction machine code format is provided, which is a **special version (different from common used version)**. The difference caused different datapath. The instruction format is shown below:

#### Instruction Machine Code Format

Instruction Type	Instruction Format							
R	Opcode [31:27]	\$rd \$rs [26:22] [21:		17]	\$rt [16:12]	shamt [11:7]	ALU op [6:2]	Zeroes [1:0]
I	Opcode \$rd [26:22]		2]	\$rs [21:17]		Immediate (N) [16:0]		

## Implementation

Below is our circuit design, including the Imem, Dmem, Regfile and Processor.



We used **Rdst** to control the instruction type (only for these instructions). Moreover, we desinged **an exception control unit** for potential exception brought by three instructions: **add**, **sub**, **addi**. We also introduced **an exception control signal**, called Exce (labled in the diagram), to deal with overflow cases.

## **Test**

We tested our design and it worked well under 50MHz.

We used waveform to test using provided assembly code. The result is as below:

